

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS ✓
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

METHOD AND APPARATUS FOR CO-SOCKET TELEPHONY

Background of the Invention

5 Field of the Invention

This invention relates generally to the integration of telephone and internet services and protocols. More particularly, the invention relates to a system whereby an internet co-socket may be associated with a standard telephone call.

Description of the Related Art

10 Call set-up technology used to establish telephone connections across the public switched telephone network (PSTN) is well known; several call set-up strategies for Internet telephony are emerging. Multimedia telephony is also emerging whereby various types of multimedia calls may be established to carry voice, video, and data. Multimedia calls are traditionally expensive and often consume large amounts of
15 bandwidth, especially if real-time video is involved. Multimedia calls are also traditionally more complex to establish and often require technical support personnel to run specialized multimedia telephony equipment. One form of multimedia call which does not require extra telephone bandwidth is a voice-over-data modem link. Voice-over-data modems allow voice to be compressed and routed across a modem connection along
20 with data. While being economical, these types of calls are still much more tedious to set up than an ordinary direct-dial telephone call. Simple multimedia call set-up strategies are provided on mixed-media packet networks such as those employing asynchronous transfer mode (ATM) and internet protocol (IP) technologies. Methods and apparatus are still needed to provide simple and economical forms of multimedia telephony which
25 allow users to transparently set up multimedia calls involving both the public switched telephone network (PSTN) and mixed-media packet switched networks.

Computer telephony integration (CTI) is also a well known and rapidly advancing technology. Examples of CTI systems include interactive voice response (IVR) call

centers whereby callers call in on a phone line and respond to digitized voice menu prompts with dual tone multifrequency (DTMF) signals (i.e., "touch tones"). The call center's IVR computer decodes the touch tone values and either provides information or routes a call accordingly. Some call centers use speech recognition in lieu of, or in addition to, touch tones. In many systems a caller can be identified using call line identification (CLID) information which is commonly known as "Caller-ID." CLID information may be derived from automatic number identification (ANI) information used to track billing in a central office switch. Signaling system number seven (SS7) links carry CLID and/or ANI information across a PSTN. SS7 call set-up information is carried on a common signaling channel separate from channels used to carry voice traffic.

The PSTN is the traditional telephone network made up of local-exchange carriers (LECs), competitive local exchange carriers (CLECs) and long distance inter-exchange carriers (IXCs). With the recent advent of internet telephony gateway servers, some PSTN calls may be partially carried over an internet to avoid tolls. For the purposes of the discussion herein, calls originating or terminating in the PSTN but partially routing over an internet via a gateway server are still considered to be PSTN calls. It is recognized that certain elements of the PSTN network may adopt packet switched techniques similar to an internet. For the purposes of the discussion herein, calls which represent plain old telephone service (POTS) and integrated services digital network (ISDN) but are carried across a packet switched IXC or LEC are also considered PSTN calls. For the purpose of brevity, a campus call which uses POTS or ISDN over standard telephone wiring and is switched by a PBX is considered a PSTN call. A call which originates using a packet switched protocol such as H.323 or other form of native multimedia packetized call is not considered to be a PSTN call.

In the present disclosure, a distinction is made between "an internet" and "the Internet". The term "internet" (lower case) is meant to apply broadly to any type of mixed media packet switched network. For example, an internet may be a local area network (LAN), a metropolitan area network (MAN), a wide area network (WAN), an enterprise network or a modernized public network. In most present day situations, networks of different types are joined by what is properly termed an internet. Hence an



internet is a network of two or more networks joined together, normally by a bridge, router, or switch. Within the present disclosure, an "internet" refers to a network which may be joined to another network using a properly selected bridge, router, or switch. A network which is in fact isolated, for the purposes of the discussion herein, may also be defined as an internet. Conversely, the term "Internet" (upper case) refers to the ubiquitous world wide web (WWW). In most cases, a given internet is connected to and hence is a part of the Internet. Some other types of internets which may not be connected to the Internet include, for example, various enterprise WANs and some cable television (CATV) networks. Hence, "the Internet" is but one example of "an internet", and thus all uses of the word "internet" herein shall apply directly to, but are not limited to, "the Internet."

There are a wide variety of known CTI systems. One system is called "screen pop." Screen pop systems recognize incoming phone numbers using CLID type information and display a screen containing information about the caller to an agent. For example, screen pop may provide an agent with a screen of information including the caller's name, personal data, buying habits, and needs. The CLID information may also be used by CTI switching apparatus to route an incoming call to an appropriate agent based on the caller's profile. These features enhance customer satisfaction and eliminate the need to ask the caller for information which may already be in an available database. Another CTI system is called screen transfer. When a call needs to be transferred from a first agent to a second agent, the CTI information screens and data associated with the call may be transferred to the second agent. Another form of CTI involves outbound dialing. An outbound dialer is a computer telephony device which dials telephone numbers to initiate telephone calls.

In call center applications, a caller navigates the IVR menu system to select an area of interest so the call can be routed to an appropriate agent. Calls to call center oriented IVRs often require the caller to wait in a waiting queue until an agent becomes available to accept the call. In most cases the caller may wait several minutes before an agent becomes available. In more extreme cases the caller may need to wait as many as

fifteen minutes, for example. An average call to an IVR call center typically involves too long of a wait in a queue. Beside offering poor customer service, long waiting queues cost companies with IVR call centers significant amounts of money. If a company has a 1-800/888 number, the company will often pay on the order of twenty cents per minute for the caller to wait in the queue. Also, the longer the queue, the more active calls the call center will need to support at a given time. This means the call center must purchase more local loop telephone lines each month from the LEC. For example, a relatively small call center with ten to twenty operators may require the equivalent of 200 or more phone lines to be able to accommodate peak traffic loads without returning busy signals. Aggregated over the United states, millions of dollars per day are spent by call centers to pay for the toll charges of callers navigating IVR systems and waiting in queues. Millions of more dollars are spent each month on the local lines needed to accommodate these callers waiting in the queues.

Another related problem is faced by organizations not having 1-800/888 numbers.

If a caller makes an out-of-pocket toll call and has to wait in a queue more than a few minutes, the caller will often terminate the call. Hence the recipient organization will lose the benefit of the call, which in the commercial context may equate to significant lost revenue.

Several prior art solutions have been proposed to deal with the foregoing problems. One such system is known as "call-back." Here a caller calls the call center, navigates an IVR menu system, and is offered the option to hang up so the IVR system can call back in an estimated number of minutes. The caller is entered into a call queue and the connection is dropped. When the caller's time arrives, the system will call the caller back, thus saving the expense of leaving the line busy while waiting in the queue.

This solution is attractive because the caller can be freed to do other tasks while waiting for the call-back. This increases customer satisfaction while at the same time the company handling the call saves money on telephone line and toll charges. One difficulty implicit in this approach is customer trust. Many call centers use variants of this approach where the caller is not called back in a timely fashion, often not even in the

same day. Hence callers are often untrusting of the system and reluctant to hang up. Accordingly, such systems typically enjoy only limited use.

Another approach to solving the telephone queue problem is through the use an Internet call center such as the Internet Call Center recently announced by Lucent Technologies Inc. of Murray Hill, NJ. In this approach a customer does not make a phone call but accesses a company's call center through the Internet. A standard web browser is used to access the Internet call center by either keying in a URL, finding Internet call center using a search engine, or clicking on a browser bookmark which references the Internet call center. In the Lucent system, an IVR system is replaced by a set of Internet browser-related dialog forms. Hence the user accesses the call center via the Internet, navigates a set of menus, and then, if needed, clicks on a button displayed in the browser with a mouse which requests an agent to call back through the Internet using Internet telephony voice packetization methods. In this way, the user is called back through the Internet and is able to talk to an agent using a multimedia PC. All of the CTI related features such as screen pop and screen transfer are available just as though the user had entered the system using a regular telephone. This system has the advantage that no telephone charges are incurred, all data and voice traffic occurs through the Internet, saving a significant amount of money for the company operating the Internet Call Center. In addition, the Internet Call center is compatible with a normal 1-800/888 call center, so agents may interact with callers entering the system through the normal phone lines or via the Internet. In either case, the same types of screen pop and screen transfer technologies may be used. A disadvantage with using this system relates to user access. An Internet call center is designed primarily to attract web users. Many users may find it difficult or tedious to find a given web site. Users may desire to call a company over the telephone network as is common practice rather than search for a web site. Hence an innovative way is needed to obtain the benefits of an Internet call center technology while providing a simple telephone access technique.

Another known technology is called "web call-back." This is a newer version of normal PSTN call-back as discussed above. In this technology, a user navigates the



Internet to reach a destination web call center. The user manipulates Internet screens using a web browser instead of a set of IVR menus, similarly to the Lucent system previously described. In web call-back, when the user selects a button, he/she will receive a call back via the PSTN from a call center associated with the Internet site.

5 Since the call-back is placed over the PSTN, the resulting connection maintains quality and a greater degree of security. Some web call-back systems display a timer on a client web browser indicative of when the PSTN telephone call-back may be expected. Other systems are not time oriented, and may require the user to wait several days before being called back. One problem with this technique is that the user must be able to locate the
10 associated web site. Often a user will know a 1-800/888 number and will find it convenient to call such a number instead of navigating the Internet. In situations involving local entities such as restaurants and movie theaters, the associated phone number may be known while it is less clear how to find the entity on the web. This problem is most pronounced because smaller local entities do not generally have easily
15 identified domain names and addresses. Another problem with current web call-back systems is that the timer only provides knowledge of when a call will arrive. It does not give the user the ability to interact with the scheduling of the return call which may be advantageous as will be described with respect to an aspect of the invention described herein below. Hence while web call-back has definite merits, it still has shortcomings
20 and limitations which need to be overcome.

Additional services are emerging which provide "web dial tone" to H.323 IP telephone users. Note "IP" stands for "internet protocol," and H.323 represents a packetized IP telephony protocol whereby voice and video telephone calls may be transmitted and received over an internet such as the Internet. Callers may initiate H.323
25 phone calls by dialing a web based telephone number. H.323 call centers may respond by providing video of a live agent. Also, a form of screen pop known as "data conferencing" may also be employed. Here the agent can see screens synchronized with a user's web browser and can "push" webpages and the like to a caller's browser. Similarly, peer-to-peer data conferences may be established for multimedia communication between
30 colleagues. A data conference is a real-time multimedia telephony session whereby two

or more user's may share images, computer screens, documents and other similar information while also being able to talk and exchange video. While existing systems allow a PSTN caller to patch into a data conference, the PSTN caller is limited to voice. A system is therefore needed to allow PSTN callers to enjoy toll quality voice and to set up data conferencing services with a computer attached to an internet.

The Internet is rapidly evolving to provide new means of access. For example, so-called "Web-TV" is a technology whereby inexpensive Internet appliances may be installed and operated using the television (TV) set as a display monitor while a CATV network provides Internet service. In these systems, the cable system provides the link layer interface to the user, and hence home Web-TV users will have a telephone free to make and receive PSTN calls while the CATV network provides internet service. A link layer interface is a signaling protocol combined with a physical channel and is used to carry data between stations. Also, as asymmetric digital subscriber line (ADSL) and related DSL technologies become prevalent, users will be able to support regular phone calls and extra digital services on the same twisted pair telephone line. Thus DSL subscribers will also be able to leave a telephone line free while being connected to the internet. Office workers and growing numbers of home users already have access to both a telephone line and a separate internet connection. Thus many users will use the Internet for packet transfers and a telephone line for toll quality circuit switched voice traffic. As Internet bandwidth increases and delays decrease, better quality Internet telephony voice circuits will also become available. This will allow user's to both place internet telephony calls and access traditional internet services using a single internet connection.

In U.S. patent 5,724,412 (hereinafter "the '412 patent"), a method and apparatus are disclosed which allows internet data to be appended to CLID packets. This patent discloses a message structure to allow callers to send a called party a message, to include a multimedia message via a CLID type packet. The invention disclosed in the '412 patent relates primarily to allowing a called party to receive a CLID notification which permits the called party to contact the caller over the internet. The patent envisions sending a screen of information, an e-mail address, a uniform resource locator (URL) to a home page, an FTP site, or other information. The concept enables a form of mixed

PSTN and internet messaging from the caller to the call recipient. However, the invention set forth in the '412 patent does not provide a means to allow such information to be used 1) to save toll charges related to call centers and IVR systems; 2) to enable more efficient web call-back systems; or 3) to provide real-time multimedia phone calls.

- 5 Hence a more capable approach is needed to send information via new forms of CLID packets which can address these problems.

Based on the foregoing, it would be desirable to have a system which integrates aspects of computer telephony with Internet services and Internet telephony. Such a system ideally would retain aspects of known CTI systems such as call-back and web
10 call-back to reduce telephone toll charges while at the same time improving customer service. It would also be desirable if a caller could access a call center using a PSTN telephone number such as a 1-800/888 or a local number, and would be able to transfer a PSTN call to an internet session.

Similarly, it would be useful to allow a caller to call a PSTN telephone number
15 and subsequently use an internet browser to perform those actions or selections currently performed using IVR. Such an integrated PSTN/internet capability would avoid the need for the caller to wait on hold on a telephone line while increasing customer satisfaction and reducing line charges and 1-800/888 related toll charges.

Another desirable feature lacking in prior art systems relates to allowing a caller
20 and an agent to converse using a toll-quality voice connection. This type of arrangement permits business transactions to occur while also providing the ability to jointly view information provided by screen pop, screen transfer, database access, and related CTI features. Additionally, providing a caller waiting in a telephone queue the ability to perform other tasks (including making other telephone calls and waiting in multiple call
25 queues simultaneously). would be of great utility

The foregoing concept of automatically setting up an internet session in response to a PSTN telephone call would also be useful to enable various forms of multimedia communication. A telephone connection is established between a caller and a callee through the PSTN or other form of telecommunications network in response to a call

placed by the caller. It would be useful to automatically provide a shared-screen of information visible on computer screens to both the caller and the callee on both sides of the telephone connection. Such communication would also include the ability to converse and share information over an inventive link which enables full service data conferencing to become automatically associated with a PSTN telephone call.

It would further be desirable to support a voice-over-data type service without the need to make point-to-point modem calls, and provide the capability to support multimedia data conferencing services associated with PSTN calls.

Summary of the Invention

The present invention satisfies the foregoing needs by providing systems and methods which allow PSTN telephone users to make use of a mixed-media packet switched internet to support a call. The present invention centers around the concept of a co-socket. A co-socket is established in support of a telephone connection. In a preferred embodiment, a caller dials a callee to establish a point-to-point telephone connection. This established telephone connection is used to carry a SYN segment which is a part of a SYN sequence to establish an internet socket connection in support of the call. When the caller calls the callee, the callee's computer preferably pops a screen of information directly on the caller's computer screen. Using Caller-ID, the callee can identify the caller and thus establish internet communication with the caller without picking up the telephone. Alternatively, the caller can pop a screen of information on the callee's computer screen. A call queue management method is presented which exploits an established co-socket. Apparatus is presented for smart phones and CTI servers which support co-socket telephony as defined herein. Methods for use of the smart telephone and CTI server apparatus are presented to set up co-sockets in support of telephone calls. Computer software structures are presented which allow the apparatus to practice the methods. Also, computer hardware structures are presented which allow apparatus to run software to support co-socket telephony methods. Finally, apparatus is presented for internet and telephone network equipment to support aspects of co-socket telephony. The

present invention includes various aspects which support co-socket telephony. These aspects are summarized immediately below.

A first aspect of the present invention involves a computer telephony server which supports co-socket telephony. The computer telephony server includes a first coupling operably connected to a telephone line so as to receive information indicative of at least one internet address associated with a caller on the telephone line. The server also includes a second coupling to a protocol stack. The protocol stack is operably connected to a link layer interface. The server also includes a computer device operably connected to the first and second couplings. The computer device is operative to route at least one data packet via the second coupling. The link layer interface is operably connected to an internet and the data packet is addressed so as to be routed to the at least one internet address in the internet. Another similar computer telephony server is also taught which is similar to the one discussed above but which includes a computer device which maintains a call queue. A related server method employing a database translation is provided as well.

In the description above the phrase "operably connected" implies a structural relationship but not necessarily a direct connection. For example, if two modules are operably coupled, they may be indirectly connected via one or more intermediary modules. Also, an "operable connection" may involve a relationship such as a connection between a CPU and a software module which runs on a CPU. It may also involve a software connection between software modules. The phrase "caller on the telephone line" refers to the caller who placed the call received on the telephone line. Moreover, a "module" is a computer device which may comprise hardware and/or software. A "computer module" is a functional block which is embodied as a logic circuit controlled by software or a hard-wired sequential logic.

A second aspect of the present invention involves a method of managing a call queue. The method is provided for use in a computer telephony server employing co-socket telephony. A first step of the method involves reading data from and writing data to a coupling to a protocol stack. The protocol stack is coupled to a link layer interface, and the link layer interface is coupled to an internet. A second step of the method

involves accepting inputs from the protocol stack indicative of selections made by a remote user. A third step of the method involves transmitting a data value via the protocol stack indicative of when a response can be expected from the computer telephony server to the caller. A fourth step of the method involves maintaining a call queue, whereby information received via the protocol stack from the caller may be used to alter the priority of a caller within the call queue. A fifth step of the method involves dialing a telephone number to establish a telephone connection with the caller when the caller's priority in the queue has reached a specific value.

A third aspect of the present invention is a method of establishing a co-socket connection. This method is practiced by equipment which initiates a connection. Equipment which initiates a connection is said to be at the "requesting end" of the connection. In a client-server paradigm, the requesting end corresponds to the client. A first step of the method involves sending a data segment from a requesting end to a telephony interface. The data segment is transmitted from the telephony interface to a remote computer via a telephone connection to initiate the establishment of a co-socket. A second step involves communicating via the co-socket with the remote computer via a link layer interface different from the telephone connection. In this method, the remote computer may be a CTI server, a peer smart telephone, or any other computerized device capable of a call via a telecommunications network. While the aforementioned method is practiced on a requesting end of a connection, a similar method is disclosed for use on the server end of the same connection. Another similar method with a slightly varying scope is also presented for use on the server end of the connection.

Another aspect of the present invention involves a smart telephone. The smart telephone is analogous to the aforementioned CTI server apparatus, but involves a client or requesting end of the connection. The smart telephone practices the method for establishing a co-socket as discussed above. The smart telephone consists of a computer telephony interface which is operative to initiate a telephone connection. The smart telephone also has a dialer operative to dial a telephone number to initiate the establishment of the telephone connection. Additionally, the smart telephone has a module which initiates the establishment of a co-socket with a remote device by

transmitting a data segment via the telephone connection. Subsequent communication using the co-socket is coupled via a link layer interface other than the telephone connection. Another version of a smart telephone is also presented which uses a database translation to determine a co-socket address of a caller. An application program for execution on the database version of the smart telephone is also presented. An internet database server used to provide co-socket addresses for use with the database version of the smart telephone is taught as well.

A fifth aspect of the present invention centers around a sockets-telephony API software library. The sockets-telephony API software library includes a co-socket connection establishment function. The co-socket connection establishment function involves a first software module coupled to a telephone connection. The first software module is operative to direct information to be transmitted and/or received via the telephone connection. The co-socket establishment function also includes a second software module coupled to the first software module and coupled to a co-socket data structure which is visible to the function. The second software module is operative to communicate with a remote computer by transmitting and/or receiving at least one data segment in a co-socket establishment sequence. The second software module causes the first software module to be run so the at least one data segment is routed via the telephone connection. Once the co-socket connection is established, subsequent communication proceeds between a process owning the co-socket data structure and a process located on the remote computer via a link layer interface other than the telephone connection. Also presented is an application program which calls the foregoing function. Also presented is an operating system including the foregoing function. In addition, a computer or smart telephone running the operating system with the foregoing function is disclosed.

A sixth aspect of the present invention involves a computer program. The program includes a coupling used to establish a PSTN telephone connection via a computer telephony interface API. The program also includes a coupling to a network via a network interface API. A software module operative to initiate a point-to-point PSTN telephone connection to a remote station using the computer telephony API is included in the program. The computer program also has a software module operative to

send a SYN segment to the remote station via the point-to-point PSTN telephone connection using the computer telephony interface API to establish a co-socket. The program also has a software module operative to accept information from a local data buffer, perform application layer formatting of the data, and transmit the formatted application layer data to the remote station via the network interface API function call.

Another aspect of the present invention involves a method of sharing information with a remote computer for use in a computer operating system. A first step of the method involves intercepting information contained within an information stream transmitted from a first local process to a second local process. A second step of the method deals with making a copy of the information. A third step involves allowing the original information stream to reach the second local process. A fourth step involves passing at least some of the copied information to a protocol stack process which in turn forwards the information to the remote computer via a co-socket associated with a PSTN telephone call.

An eighth aspect of the present invention deals with apparatus coupled to a telephone switch. The apparatus employs a translation unit which receives ANI or CLID data and translates said information relating to a caller's client internet socket address. The apparatus also employs a module which places information relating to the caller's internet socket address into a data packet for transmission to a dialed telephone number. The internet socket address includes a port number to enable a callee to send a screen of information via an internet to be displayed on the caller's computer screen. A apparatus similar is also disclosed which evaluates dialed number information and ANI or CLID information and generates a packet sent via an internet to establish a co-socket in support of a call. A related method of processing telephone calls within a telephone network in support of co-socket telephony which may be practiced by the disclosed apparatus is also presented.

Brief Description of the Figures

The various novel features of the invention are illustrated in the figures listed below and described in the detailed description which follows.

5 Figure 1 is a block diagram representing a CTI server and a smart telephone connected via a point-to-point PSTN connection and an internet socket connection according to the present invention.

Figure 2 is a flow chart illustrating a CTI server web call-back processing method according to the present invention.

10 Figure 3 is an exemplary information window display used to assist a user in managing multiple telephone queues.

Figure 4 is a functional block diagram illustrating a point-to-point PSTN connection/co-socket architecture using a single DSL or multiplexed telephone line.

Figure 5 is a flow chart illustrating a method of co-socket initiation utilized by a calling device according to the present invention.

15 Figure 6 is a flow chart illustrating a complementary method of processing utilized by a called device to respond to and participate in the establishment of a co-socket in accordance with the method of Figure 5.

Figure 7 is a block diagram illustrating an application programmer interface used to interface to a protocol stack having co-socket establishment capabilities.

20 Figure 7a is a block diagram illustrating one embodiment of a software architecture for use on a smart telephone or computer according to the present invention.

Figure 8 is a block diagram illustrating a filter program according to the present invention which is operative to intercept bytes within an information stream and transmit a copy of the intercepted bytes across a socket or co-socket.

25 Figure 9 is a flow chart illustrating a method of sharing screen information with a remote computer using a filter module.

Figure 10 is a block diagram of one embodiment of a central office switching apparatus according to the present invention, which is used to perform ANI to co-socket address database translations.

Figure 11 is a flow chart illustrating a method of performing caller identification to co-socket address database translations.

Figure 12 is a block diagram of an inter-exchange carrier network architecture used to perform ANI to co-socket address database translations according to the present invention.

Figure 13 is a flow chart illustrating a method of initiating the establishment of a co-socket by a calling device using an internet database approach.

Figure 14 is a flow chart illustrating a method of removing internet address data from PSTN data packets according to the present invention.

Detailed Description of the Invention

Reference is now made to the drawings wherein like numerals refer to like parts throughout.

Figure 1 illustrates a first embodiment of a communication configuration whereby a CTI server 100 and a smart telephone 115 are each coupled to a PSTN 105 and an internet 110. A peer device 102 may be another device similar to the smart telephone 115, and is also coupled to the PSTN 105 and the internet 110. An optional database system 112 may also be coupled to the network configuration via the internet 110. As will be discussed in connection with the embodiments illustrated by Figures 10-12, the database 112 may also be implemented by equipment within the PSTN 105. The PSTN 105 may be coupled to the internet 110 via various types of connections 107 which may include a gateway connection to route long distance voice traffic across the internet 110.

In a preferred embodiment, the database module is an internet server. The database module 112 includes a coupling to the internet 110. This coupling is operative to receive a client request from the internet 110 including information relating to a

telephone number associated with a potential callee. The database 112 associates the telephone number with an internet address. The database 112 includes a reply module connected to the coupling and is operative to return a data packet via the coupling to the internet 112 upon association of the request with the internet address. The data packet
5 sent out by the database 112 is related to the internet address of the callee. Once this data packet is returned to the smart telephone 115, the smart telephone 115 may initiate a co-socket with the callee instead of, or in addition to a telephone connection. Likewise, the CTI server 100 may read CLID data associated with an incoming call and send this information to the database 112 to find out an internet address related to the caller. This
10 allows the CTI server 100 to pop a screen of information on the caller's screen in response to an incoming call. In an alternative embodiment, the CTI server and/or the smart telephone 115 may incorporate a local database to translate telephone numbers to internet socket addresses.

The CTI server 100 includes a main computer 120 which is coupled to a CTI
15 interface 125, an internet protocol stack 130, a storage unit 135, and an optional call center 140. For example, the main computer 120 may be implemented with an Intel Pentium™ processor or a computer board such as a mother board built around an Intel Pentium™ processor. In this case the coupling to the storage area 135 may be implemented using a bus connection to a solid state memory. In most embodiments, the
20 storage area 135 is implemented at least partially with a mass storage device such as a disk drive. In this case, the mass storage unit 135 is preferably coupled via a disk controller which is itself coupled to the main computer 120 via a standard type of bus interface such a peripheral component interconnect (PCI) bus. The storage unit 135 may be configured to support voice mail and other forms of multimedia message storage. The
25 internet protocol stack 130 is typically implemented as software process on the main computer 120 and hence is coupled via a software connection such as an interprocess communications channel.

Application programs running on the main computer 120 typically access the internet protocol stack 130 using a function call to an API such as a WinSock API or a
30 UNIX™ Sockets API. The CTI interface 125 is also usually implemented as one or more

software processes which execute on the main computer 120. An application program running on the main computer 120 is coupled to the CTI interface 125 via an interprocess communications link provided by an operating system running on the main computer 120. In this case, the coupling is provided between an application program running the
5 main computer 120 and the CTI interface 125 via a call to a CTI API. In general, an argument list to a function call typically coupled information from the main computer 120 to the CTI interface 125. Call queue and IVR systems are ideally maintained by applications programs which run on the computer 120. The call center 140 preferably consists of an automatic call distribution (ACD) system which routes incoming calls to
10 one or more call processing agents. The main computer 120 is typically coupled to the call center 140 using a plurality of twisted pair cables, one pair for each agent station. In some embodiments trunking may be employed whereby a single twisted pair carries a plurality of calls to a remote call center and multiplexing and demultiplexing equipment is used to separate and combine the multiple calls on the single twisted pair. Some
15 systems may couple the call center 140 to the main computer 120 using LAN technology. LAN technology for call distribution involves compressing and packetizing calls for shipment over a coaxial cable. An optional data link 145 couples the CTI interface 125 directly to the internet protocol stack 130, which is of the type well known in the art. This direct data link 145 allows information at lower layers (network, link and physical
20 layers) of the protocol stack 130 to communicate with the CTI interface 125. This data link is typically embodied either as a parameter list involved in a function call or via an interprocess communication mechanism.

Couplings between all the modules in the CTI server may be implemented through the computer 120. For example, these couplings may be implemented by
25 auxiliary data paths, direct memory access transfers, or directly by the processor using move operations. The input/output structure of the internet protocol stack 130 preferably includes a first coupling to an application layer via an API accessed by an application program running on the computer 120, and a link layer interface to a remote network device in the internet 110. Common examples of link layer interfaces include modems
30 which connect a home to an ISP, a PSTN voice telephone circuit established between two

users, a CATV connection from a Web-TV client to a CATV based ISP, or a LAN connecting from a computer to a remote internet server. Each of the foregoing link layer interfaces (as well as other similar arrangements) may be used within the present invention.

5 Referring again to Figure 1, the smart telephone 115 includes several subcomponents. The smart telephone is controlled by a processor 150 which can be any logic processor including for example an ASIC, an embedded microprocessor, or computer motherboard, each of which are well known in the art. For example, an Intel Pentium™ processor may be used in a smart telephone implemented as a personal
10 computer. The Intel Pentium™ processor employs on-board functional units and an on-board cache to execute a version of the so-called Intel Instruction-Set. The Intel Pentium™ interfaces to an external system using a data bus, an address bus, and a control bus. Stand-alone smart phones which are not implemented as a part of a personal computer may be implemented with specialized microprocessors such as the ARM
15 processor from Advanced RISC Machines Inc. The ARM processor employs a reduced instruction set computer (RISC) architecture and may be integrated on the same chip with special purpose circuitry to form a customized type of microcontroller. More standard types of microcontrollers such as the MC68HC16 by Motorola Inc. may be also be used to implement the processor 150 of the smart telephone. The MC68HC16 integrates a 16-
20 bit MC68000 processor core by Motorola Inc. with a collection of peripheral devices on the same chip. The interface and design techniques for all of these types of processors are well known and are documented in the microprocessor literature and the documentation available from the aforementioned companies. Once programmed, the processor 150 may be viewed as being constructed of various computer modules which interact and perform
25 their programmed functions. In an ASIC design, the processor 150 may be implemented as an interconnected combination of hardwired computer modules.

The processor 150 is coupled to a CTI interface 155 and to an internet protocol stack 160. The CTI interface is also preferably coupled to a handset 158 which may alternatively be coupled directly to the processor 150. The internet protocol stack 160
30 ideally has a structure similar to the protocol stack 130 as discussed in connection with

the CTI server 100 above. An optional direct coupling 163 may be provided between the CTI interface 155 and the protocol stack 160. The processor 150 is preferably also coupled to a storage unit 165 which includes memory and which may include mass storage implemented in technologies such as hard disk drives, solid state disk drives, or other means. The processor 150 is also coupled to an input/output terminal 170 which includes a graphics display capable of displaying windows and/or dialog boxes. The input/output terminal also preferably includes a keyboard and/or mouse to accept user inputs. In the present embodiment, the processor 150 is also be advantageously coupled to a one or more sensors 175. Examples of sensors which may be coupled to the processor include a video camera, a digital camera, a scanner or other multimedia data collection means. In a second embodiment of the smart telephone of the present invention, the processor 150 is also coupled to a wireless link layer interface 180. The wireless interface 180 may involve a wireless LAN and is preferably used to couple the smart telephone 115 to a remote unit 185. The remote unit 185 provides mobility to the user by allowing him or her to leave the vicinity of the smart telephone but still be able to handle calls and manage program operation on the smart telephone.

The apparatus of Figure 1 may operate in many different modes to support various methods and applications. Consider first a scenario whereby a caller using the smart telephone 115 wishes to call the call center 140 which is coupled via the CTI server 100.

In this case the smart telephone 100 is associated with the caller. Assume that the call center 140 is reachable via a PSTN 1-800/888 number. A caller preferably dials the 1-800/888 number using either a keypad associated with the handset 158 or an automatic CTI-controlled dialer which may be part of CTI interface 155. In this case the phone number is found in a list or is entered using the I/O windows interface 170. Whichever dialing method is selected, the dialed 1-800/888 number transmitted over a telephone line to a PSTN switch located in PSTN 105, possibly via a PBX (not shown). At this point the PSTN may interact with the dialed number or the CTI interface may add extra data to the dialed number indicative of an internet socket address. For now, suppose no extra information is added, but the PSTN 105 routes a PSTN data packet to the CTI interface 125. A PSTN data packet is defined as any data packet which routes over the PSTN.

PSTN data packets are used for call set-up, communication systems management, and other purposes. PSTN data packets are most typically sent using SS7 or X.25 protocols.

When the PSTN data packet arrives at the CTI interface 125, several actions may occur. For the present example, consider the case where the PSTN data packet includes only standard CLID information to identify the caller. In one embodiment of the present invention, the CTI interface 125 controls the line pick-up process to accept the call. At this point CTI interface 125 listens for a tone sequence. In this exemplary embodiment, the expected tone sequence contains a synchronize sequence numbers (SYN) segment used to set up an internet connection. SYN segments, and internet stream socket and datagram socket connection establishment procedures are well known in the prior art, and are discussed in full detail in "TCP/IP Illustrated Vol. 1-3," by W.R. Stevens Addison-Wesely Publishing 1994, which is incorporated by reference herein in its entirety. A segment is generally defined in TCP as a timed packet. When a segment is transmitted, a timer starts, and when an acknowledgment is received, the timer stops. If the timer runs out, a new segment is transmitted. In the TCP packet header there is a flag called SYN. This flag is set during the exchange of segments used to establish a TCP connection. A TCP connection is also called a stream socket. The SYN packet exchange sequences used to set up a TCP connection are discussed in detail in Section 18.6 of Volume 1 of the aforementioned Stevens reference. SYN segments are embodied as UDP datagrams in the TCP protocol. Note that while TCP is chosen as the protocol for use with the present embodiment, other protocols and formats may be used with equal success. For example, a "co-socket" may be established using a session layer protocol which executes over a transport layer protocol. Standard sockets are transport layer connections. Co-sockets as defined within the scope of the present invention may be defined at any specified layer of any selected protocol stack without restriction. Hence at any point, a SYN segment may be replaced with a connection establishment packet for a given protocol.

Variations of the TCP connection establishment procedure may be desirable when carried out wholly or partially across a point-to-point PSTN telephone connection instead of an internet. Therefore, as used herein, a "SYN segment" is generally defined as any

data which may be transmitted via various methods and formats as a part of any internet connection establishment process. In one generalized form, a SYN segment may simply constitute an internet address used to indicate an internet address and port number needed to begin a TCP/IP SYN segment exchange sequence. Also, as used herein, a point-to-
5 point telephone connection is generally defined as any dialed telephone connection set up between two or more telephones. Furthermore, a point-to-point PSTN telephone call is defined as any telephone connection between users set up using a PSTN or PBX originating or terminating line. This connection may be established via a PBX between phones within an enterprise, and may also route partially across an internet. The point-
10 to-point PSTN telephone call is not established as a PC-to-PC multimedia packetized telephone call using a format such as H.323. Such telephone calls already have full multimedia capabilities.

Referring again to the exemplary embodiment of Figure 1, the CTI interface 125 answers the phone in response to a ring-signal and listens for a tone sequence. If the
15 caller is using a smart telephone such as the smart telephone 115, the smart telephone 115 is the requesting end of the connection so transmits a SYN packet over the established point-to-point telephone connection. If the phone goes unanswered, but CLID or related information is accepted by the called number, then a point-to-point connection is still established in response to the call, but the point-to-point connection may be termed a
20 PSTN-datagram connection instead of a voice circuit. Again turning to the aforementioned example, the CTI interface receives and decodes a sequence of tones indicative of a SYN segment. In this example the SYN packet is transmitted from CTI interface 155 to CTI interface 125 and is used to set up a TCP/IP stream socket connection across the Internet. Hence in this example, internet 110 corresponds to the
25 Internet.

Several methods may be used by the apparatus of Figure 1 to establish an internet connection using a point-to-point telephone connection which has been established via the PSTN 105. These methods are discussed in greater detail below in connection with Figures 5-7 and Figure 11. For present, assume that an entire TCP/IP stream socket
30 connection establishment SYN sequence is exchanged between smart telephone 115 and

the CTI server 100 using the CTI interface 155 and the CTI interface 125. The CTI interface 125 is coupled to the protocol stack 130 via the computer 120 or the optional direct coupling 145. Likewise, The CTI interface 155 is coupled to the protocol stack 160 via the processor 150 or the optional direct coupling 163. Using the apparatus and method described thus far, in this example a TCP/IP stream socket is established via the CTI interfaces 125 and 155. The connection is then passed across to the protocol stacks 130, 160 to enable internet data communication. The direct couplings 145 and 163 are provided to allow the protocol stack processes 130, 160 to directly interact using the CTI interfaces 125, 155 which communicate using the point-to-point telephone connection established across the PSTN 105 as a link layer. Once connection establishment data has been sent, the protocol stack may then switch its link layer interface from the point-to-point PSTN link to a link coupled to the internet 110. Note a stream socket established in this manner is a co-socket as defined hereinabove.

As will be discussed in greater detail below, many variations of the above described method may be used to establish an internet co-socket. The above example illustrates one variation whereby the entire co-socket connection establishment sequence is carried out using the point to-point PSTN connection to supply the link layer interface. Subsequently, the link layer of the co-socket is switched a link layer interfaces coupled from the protocol stacks 120, 160 to the internet 110. Another embodiment requires that only a first SYN segment be transmitted over the point-to-point telephone link with all subsequent SYN segments being transmitted on the link coupled to the internet 110. As further discussed below, still other embodiments utilize an initial SYN segment which is encoded into a CLID packet and transmitted via a PSTN data packet such as an SS7 packet. The CLID information of the caller may be replaced by SYN data using a variety of methods as discussed herein in connection with Figures 5-7 and Figure 11.

Referring now to Figure 2, a call processing method 200 employed by the CTI server 100 when called by a smart telephone client 115 via a point-to-point PSTN connection is disclosed. For purposes of illustration, consider the example wherein a caller is required to navigate a set of menus and wait in a queue to speak to an agent residing in the call center 140. In a first step 205 the CTI server receives a SYN segment

over the PSTN. This SYN segment may be transmitted using a CLID-like packet sent via SS7 before the server 100 answers the call, or the data packet may be transmitted over the point-to-point PSTN connection using a tone sequence once it is answered by the CTI server 100. In exemplary embodiments where a non-smart phone initiates the point-to-point PSTN connection, the SYN segment may be transmitted by having the caller key in touch tones, press an automatic dialer button, or speak an internet address which is then converted using voice-to-text translation within the CTI server 100. It can be appreciated that a broad variety of different combinations may be used in such applications without departing from the spirit of the invention.

Once the SYN segment data packet is received, an internet connection is established in a second step 210. For example, suppose the internet connection is a TCP/IP stream socket set up using the co-socket techniques of the present invention. More details of how this step may be performed are provided in connection with Figures 5-7 and Figure 11. Once the internet connection is established per the second step 210, control passes to a third step 215. In this third step 215, the call may be preferably dropped or rejected. This step 215 is optional and may be initiated, for example, only after receiving a caller consent input. If the internet socket is established per the second step 210 using a data packet received in the first step 205 within a SS7 CLID packet, then the call need never be answered, but may be rejected outright in the third step 215. After or before the optional third step 215, control is passed to a fourth step 220 which transmits a dialog form to the caller via an internet. The reason the call may be dropped or rejected is that the caller may use a dialog box such as the dialog box 300 (see discussion of Figure 3 below) displayed on the computer screen 170 instead of the IVR voice prompts presented by the CTI server 100. Hence when the CTI server gets a confirmation indicating the caller's computer has cooperated in establishing a co-socket, the call is preferably rejected because the caller will use the internet call queue discussed below and the more expensive and less friendly telephone connection is not presently needed.

The dialog box 300 is maintained over the internet, ideally using a web browser. In a first embodiment, the dialog box may appear directly in a web browser window, or



alternatively may be displayed in a separate window using, for example, a CGI script, a browser plug-in, or a Java applet. CGI scripts, browser plug-ins, and Java applets are well known in the computer arts and are discussed in detail, for example, in "Netscape Technologies Developers Guide" by L. Duncan and S. Michaels, Ventana Publishing, 1997, and "Web Developer's Secrets," by Harold Davis, IDS Books Worldwide, 1997, both of which are incorporated by reference in their entirety herein. It should also be noted that any application layer program may be used to accept and display the aforementioned dialog box.

Once the dialog box 300 is displayed on the caller's computer screen based on information received from the fourth step 220, control passes to an optional fifth step 225 which accepts a remote caller's inputs across the internet, preferably by accessing a sockets API. After the optional user inputs are received, a decision 230 is made to determine if more input is needed, in which case control is passed back to the fourth step 220. The fourth step 220 transmits another dialog box and passes control to the fifth step 225 which waits for more user input. Different variations of this input scheme are known to those skilled in the art and are within the scope of the present invention. Once no more inputs are required, control is passed to a sixth step 235 which transmits one or more queue timer values to the remote caller via the internet. Depending on the application program used to display the timer at the remote location, the timer can be continually decremented and redisplayed in a count-down fashion on the display to the remote caller's terminal. If the CTI server 100 wishes to modify the estimated caller queue wait time, it sends out an updated time which will divert the decrementing counter from its path by reinitializing it to a new estimate. The sixth step 235 proceeds until either the timer runs out or additional wait time is requested by the remote caller. If additional time is requested, a decision 250 is made to accept the additional time and to pass control back to the sixth step 235 to add this time to the timer. The decision 250 also allows the user to freeze a timer which in effect, adds more time to the queue-wait. After the timer has timed out, control passes to a seventh step 240 which dials a PSTN phone number associated with the caller. The PSTN phone number may be obtained by the CTI server

100 from a CLID packet, a CGI script, a Java applet, user input to a dialog box, a database, or other well known technique.

The aforementioned method 200 is especially useful because it allows a caller to manage multiple tasks in parallel. As a first example, suppose a caller wishes to make
5 three phone calls at once, each of which involve non-trivial telephone queue waiting times. Suppose the caller initiates the calls from the smart telephone 115. The caller calls a first CTI server and establishes an internet dialog session as discussed in connection with the method 200 above. After interacting as in the fifth step 225 with the dialog box 300, the user receives an estimated queue time according to the sixth step 235.

10 This time is displayed in a first timer window. Assume for example that the queue wait is ten minutes. Next the caller decides to make a second phone call while the first timer is running. The same process is repeated, and a second queue time of twelve minutes is reported. Next the caller places the third call and repeats the process once more, this time receiving a queue wait time of six minutes. Depending on the application layer program
15 running on the smart phone 115, either multiple timer windows may be displayed or a single timer-manager window may be displayed. A timer-manager window allows a user to conveniently view multiple queue-wait times and increment times in various increments using point-and-click technology as discussed in connection with Figure 3 herein. The user may click on various buttons to freeze remote queue-timers or add time.
20 Further assume that by the time the user receives the six minute time for the third call-queue, the first call queue is reduced to five minutes. In traditional systems this would create a difficulty unless the call to the first call center requires less than a minute. By freezing timers or adding times, the caller may let the call queues run down and freeze them near the one minute point or may add time to them. Hence instead of waiting in
25 three consecutive queues, the caller can spend the time making the calls instead of waiting.

In another scenario, assume that the caller is waiting in a twenty minute call queue, and after waiting eighteen minutes, the caller's attention is distracted by an urgent matter for five minutes, or a second call is received (such as from an important customer).

30 Instead of losing the eighteen minutes accrued waiting in the queue, the caller may freeze

the timer at the eighteen minute mark until the urgent matter or second call is completed.

Upon completion, a freeze button such as the freeze button 315 of Figure 3 below may be actuated whereby the timer is again allowed to proceed. In essence, the caller has allowed others in the queue to progress ahead without having to reinitiate the call and start at the beginning of the queue again. Additionally, the CTI server associated with the call center has not accrued telephone toll charges during the "freeze" period. The ability for the user to interact with the amount of time left in the call queue represents an improvement over prior art web call-back systems as previously described.

Another beneficial aspect of the smart telephone 115 of the present invention is the ability for the caller to be physically separated from the smart phone while waiting in one or more call queues. The wireless interface 180 is used to transmit timer status information (and optionally a voice connection) to a user who is remote to the smart telephone 115. Assume for example that a caller is managing multiple call queues and is subsequently called away from the smart phone 115. The caller then directs the system to forward all the timer data and a voice circuit to the remote unit 185. By carrying the remote unit 185, the caller can be at a different location from the smart telephone and perform many or all of the functions discussed above using the smart telephone 115 via the remote unit 185.

It should be noted that the web call-back system as described above may be initiated by either PSTN calls or, for example, H.323 packet calls. In this latter case, the first step 205 of the method 200 involves receiving SYN packets over an internet instead of the PSTN. The benefits of allowing the caller to freeze queue timers or add time to call queues may thus provided to web callers as well.

A first embodiment of a user interface menu 300 displayed by a smart phone according to the present invention is illustrated in Figure 3. The user interface menu 300 may be displayed by an application program to include a web browser's main window, a plug-in, a CGI script, an applet, or any program from within an operating system. A timer window 305 may preferably display the remaining wait-queue time for each active wait-queue. An add button 310 is used to cause time to be added to the associated time as displayed by timer 305. A freeze button 315 is used to cause the associated timer 305 to

freeze, as previously described. A count button 320 is used to cause the associated timer 305 to resume counting. When these buttons are selected, packets are transmitted to the remote CTI server 100 and are acted thereupon in the eighth step 250 of method 200 as shown in Figure 2. Additional inputs and buttons may be made available to add times and perform other interactions with the wait queues using extra dialog windows or icons as illustrated by the graphics object 325. For example, a graphics object 325 may support control commands to forward data and calls to the remote unit 185. The smart telephone 115 supports other inventive uses for CTI initiated co-sockets. These uses are discussed below in connection with Figures 8 and 9. Note also that the smart telephone 115 may also be used to perform internet database dialing as discussed in connection with Figure 13.

Referring now to Figure 4, a line interface technique is illustrated for connecting the smart telephone 115 to support a point-to-point telephone connection and an internet link layer interfaces using a single telephone line as a physical layer interface. In the embodiment of Figure 4, the smart telephone 115 is coupled to a line interface 400 which preferably includes a digital subscriber line (DSL) modem of the type well known in the art. The DSL modem is operative to convert a high speed data stream, for example on the order of up to 6Mb/s into an out-of-band carrier signal used to transmit data across a phone line. Meanwhile a plain old telephone service (POTS) interface is used to support a point-to-point telephone connection to carry voice traffic. In some cases the POTS interface may be substituted for a digital interface such as an ISDN line. Alternatively, voice traffic may be multiplexed directly into the DSL data signal. In most present day DSL applications, a splitter 420 is used to separate the POTS voice circuit from the out-of-band DSL data. Alternatively a splitterless DSL modem technology may be used. The basic concept behind the splitter 420 is to provide a means to separate the voice channel from the data stream transmitted over the DSL channel. If the voice data is packetized and multiplexed onto the DSL data signal, then the splitter 420 may be viewed as the multiplexer which performs this function. In this case the splitter block 420 should be physically located to the other side of the DSL modem 415 and the POTS interface 405 so as to be connected between these devices and the smart telephone 115.



As shown in Figure 4, the splitter 420 provides a coupling to a telephone line 425. The telephone line 425 couples via local wiring or wireless means to a second splitter 430. For example, in systems employing splitters, the splitter 430 separates the PSTN voice circuit from the DSL circuit and forwards the PSTN voice frequency signal to the PSTN while forwarding the DSL signal to a DSL interface. In most cases the splitter/multiplexer 430 decode the DSL signal and sends a bit stream to an internet service provider (ISP) 440 over a digital link. For example, many DSL signals can be multiplexed onto a fiber and routed to the ISP 440. The digital data is forwarded from the ISP 440 via an internet 465 to a link layer interface 470 which carries typically a TCP/IP packet stream to a remote site 460. If the multiplexed approach is used whereby the voice circuit is multiplexed onto the DSL signal, then the splitter/multiplexer 430 will separate the voice data and forward it to a PSTN (or local) voice circuit. A PSTN interface 435 is used to process the voice circuit and to couple it to the PSTN 445. The PSTN 445 is used to couple the voice circuit to the remote site 460 over a line telephone 450, preferably controlled by a CTI interface.

The operation of the apparatus of Figure 4 is essentially the same as that discussed in connection with Figures 1-3 above. Details of the CTI based co-socket connection techniques are discussed in connection with Figures 5-6 and Figure 11 herein. Again, the concept of utilizing the smart telephone 115 having a PSTN link layer interface to establish an internet session, which is then switched over to a link layer interface connected to the internet 110, is employed. Note a telephone connection is a link layer interface which connects a caller to a callee. This functionality can in fact be implemented in separate links using a single telephone line as a physical layer interface. Additional but related embodiments involve a second telephone line, a separate LAN connection, a Web-TV internet access point or an enterprise WAN connection, for example. The use of DSL is a special case of the smart telephone configuration wherein two separate links are effectively provided by a single telephone line used as a shared physical layer interface between two link layer interfaces.

Referring now to Figure 5, a method 500 is illustrated to establish an internet co-socket with a remote station via a separate point-to-point telephone connection. This

method enables, inter alia, multimedia phone calls which route voice traffic over the PSTN, and also route data traffic over the Internet. In an exemplary embodiment, the method 500 is practiced by the smart telephone 115 of Figure 1 which represents the calling or requesting end of a connection, although other configurations can be used. In general, the method 500 may be practiced by any telephone device which initiates a multimedia co-socket connection. In the exemplary embodiment, the method 500 starts when the smart phone 115 has already dialed a phone number corresponding, for example, to the distant end CTI server 100 or the peer smart phone 102 of Figure 1. In a first step 505, a SYN segment is sent to the distant end using a CLID packet, touch tones, speech, or other available means. In campus or similar environments, a fixed or dynamic table mapping ANI data to socket addresses may be used. In this case a "dumb" telephone (i.e., a standard telephone without processing capabilities) may be used, since an enterprise computer can be used to map the dialed number and ANI data into a SYN segment containing the information needed to initiate a session to establish a co-socket between the computers belonging to the caller and the callee. Hence in this campus environment, even a "dumb" telephone can cause information to be routed directly to a callee's socket address to automatically pop a screen directly on the callee's computer. Similarly, ordinary telephone conference initiation procedures may be followed to automatically initiate co-sockets for data conferencing between multiple users without the users needing to log into a common webpage or otherwise set up the data conference. Embodiments using ANI and CLID packets are discussed in connection with Figures 10 through 12 herein. Whatever the method chosen, the first step 505 involves transmitting at least enough information to notify the distant end and prompt it to establish a co-socket.

Upon completion of the first step 505, control passes to an options decision point 510. The options decision point 510 may represent a selection operation or may represent a hard-wired transfer of control to, for example, either a second step 520, a fifth step 550, or a sixth step 560. Consider the path whereby control passes from the first step 505 to the second step 520. The second step 520 receives an acknowledgment packet over the CTI interface 155. The received packet preferably contains an internet address.

With this internet address available, control passes from the step 520 to the third step 530 which passes the received internet address to a protocol stack which in turn establishes an internet connection with the remote CTI server 100 or the remote peer 102. Ideally, the internet connection is a TCP/IP stream socket established across the Internet, although
5 other configurations are possible. Once the connection is established, control then passes to a fourth step 540 which uses the internet connection to pass data, preferably between application processes residing on the smart telephone 115 and the remote CTI server 100, or the peer 102. The established internet connection is a co-socket to the point-to-point PSTN telephone connection used to receive the server's internet address. Note that if
10 control transfers from the first step 505 to the second step 520, the first step 505 only must transmit enough information to prompt the distant end to send an internet address. It is contemplated by the present invention that in such a situation, the first step 505 could be eliminated altogether by having the distant end transmit an internet address to all callers, thereby simplifying call processing.

15 Next consider the method 500 of Figure 5 whereby control passes from the first step 505 to the fifth step 550. In the fifth step 550, an entire SYN sequence is exchanged using a point-to-point PSTN connection between the smart telephone 115 and either the remote server 100 or the peer 102. This SYN sequence is used to establish an internet connection. This internet connection, as discussed above, is a co-socket. Once the SYN
20 segment exchange is completed, control is transferred to the fourth step 540 which uses a separate link layer interface to connect to the internet. This separate link layer interface is connected to an internet and is also called a "network interface." The separate link layer interface connected to an internet is different from the telephone connection. Thus, the point-to-point telephone connection may continue to be used or may be dropped if
25 desired and communication may take place only over an internet. In the case of peer-to-peer communications, for example, it may be desirable for users to converse over the PSTN link and to exchange multimedia data over the co-socket.

Consider now the method 500 whereby control passes from the first step 505 to the sixth step 560. In the sixth step 560, at least one SYN acknowledge packet (which is
30 typically the first part of a connection establishment sequence) is received over the CTI

interface 155. If the path involving the sixth step 560 is selected, the first step 505 will preferably send the first SYN segment in the internet socket establishment sequence. Once at least one SYN acknowledgment packet is received in the sixth step 560, control passes to a seventh step 565 which continues to exchange any remaining SYN segments
5 over a separate link layer interface connected to the internet 110. Once the internet co-socket connection is established, control passes to the fourth step 540 which allows application processes residing in the smart telephone 115 and the remote CTI server 100 or peer 102 to exchange information.

Referring now to Figure 6, a method 600 is illustrated for establishing an internet
10 co-socket at a remote station via a separate point-to-point PSTN connection. This method is practiced by a receiving (i.e., callee) side of a connection such as the CTI server 100. Additionally, this method 600 interacts with the previously described method 500 (Figure 5) to enable multimedia phone calls which route voice traffic over the PSTN and data traffic over an internet. In the exemplary embodiment discussed in connection
15 with the method 500, the method 600 is practiced by the CTI sever 100 or the peer 102 upon receipt of a connection request from smart telephone 115. As part of the method 600, a first step 605 is carried out to receive the SYN segment sent from the first step 505 of the method 500 as placed by the smart telephone 115. In the first step 605, a SYN segment is received from the distant end via a CLID packet, touch tones, speech, or other
20 suitable transmission format.

Upon completion of the first step 605 of Figure 6, control passes to an options decision point 610. The options decision point 610 may represent a selection operation or may represent a hard-wired transfer of control to, for example, to either the a second step 620, a fifth step 650, or a sixth step 660. Consider the path whereby control passes from
25 the first step 605 to the second step 620. The second step 620 responds to an internet address query received in the first step 605 and transmits an acknowledgment which is preferably an internet address over the CTI interface 125. Once this internet address information has been transmitted, control passes from the second step 620 to the third step 630 which represents a listening server side internet connection preferably
30 corresponding to the transmitted internet address. The third step 630 is used to perform

the server side protocol of a client-server connection establishment exchange with the remote client who received the appropriate internet via the CTI interface. For example, the smart telephone 115 calls the CTI server 100 or the peer 102 of Figure 1 and establishes a co-socket. Once the connection is established, control then passes to a fourth step 640 which uses the internet connection to pass data, preferably between application processes residing on the CTI server 100 and smart telephone 115. If this method is practiced by a peer device 102, then a peer-to-peer connection establishment protocol may be employed similarly to a client-server protocol, or the peer may act as a server in the connection.

10 Next consider the method 600 of Figure 6 whereby control passes from the first step 605 to the fifth step 650. In the fifth step 650, an entire SYN sequence is exchanged using a point-to-point PSTN connection between the called CTI device and the calling smart telephone 115. This SYN sequence is used to establish an internet connection. This internet connection, as discussed above, may be termed a co-socket connection because it
15 requires a PSTN (or PBX) co-channel to establish it. Once the SYN sequence is completed, control is transferred to the fourth step 640 which uses a separate link which connects to the internet. Thus the point-to-point connection may continue to be used or may be dropped and communication may take place over the internet connection. In the case of peer-to-peer communications, it may be desirable for two or more parties to
20 converse over the PSTN link and to exchange images and other forms of data over the established co-socket connection.

 Consider now the method 600 of Figure 6 whereby control passes from the first step 605 to the sixth step 660. In the sixth step 660, at least one SYN acknowledge packet which is part of a connection establishment sequence is transmitted over the CTI
25 interface 125. If the sixth step 660 is selected, the first step 605 will preferably receive at least one SYN segment in an internet socket establishment sequence. Once at least SYN acknowledgment packet is sent in the sixth step 660 over the internet, control passes to a seventh step 665 which continues to exchange SYN segments over a separate link connected to the internet 110. Once the internet co-socket connection is established,
30 control passes to the fourth step 640 which allows application processes residing in the

called computer to exchange information with the application processes in the calling computer. For example, the called CTI server 100 or peer 102 may exchange information with the calling smart telephone 115.

The method 600 also contemplates an alternative co-socket establishment procedure. In the alternative procedure of the method 600, the data packet received in first step 605 is preferably a CLID data packet. The CLID data packet may comprise a caller's name and telephone number as is common in the art. Also, the received information may comprise a modified CLID packet which includes a caller's internet socket address suitable for popping a screen of information on the caller's computer screen. Apparatus to provide such an internet socket address in a CLID data packet is discussed in connection with Figures 10-13. If the CLID data packet does not include an internet socket address, then the first step 605 is operative to perform a database translation to convert the CLID information to a caller's internet socket address. The database used to perform this translation may be local or remote. That is, the step 605 may involve performing a database query to a remote database accessible through the internet. A remote database which may be used for this purpose is shown as the database 112 in Figure 1. In the alternative procedure of the method 600, control next passes from the first step 605 directly to the step 665, bypassing the optional step 660. In the step 665, an internet socket is established from a computer practicing the method 600 back to a computer controlled by the caller. The step 665 is preferably performed over a link layer interface other than the telephone connection on which the CLID data packet was received. This alternative procedure has the advantage that a caller can call from a standard telephone and still receive a screen of information from the callee. For example, a caller can call from a standard office telephone and receive a screen of information from the callee. Similarly, a caller can initiate a call from a standard telephone in a home and receive a screen of information on a computer connected to a second line, a CATV terminal, or a Web-TV internet appliance connected to a television set.

The method 500 and the methods 200, 500, and 600 may all be advantageously implemented via a computer program which executes on a processor coupled to a memory. The same holds true for the methods 900, 1100, 1300, and 1400 which will be

subsequently discussed. In most cases these methods may be implemented using an application program as defined below. A computer program is defined herein generally as a sequence of instructions which executes on a processor. Typically the computer program is stored in a semiconductor memory, and instructions are read by a processor from the memory in sequence. The processor typically executes the instructions sequentially in the order in which they are read from memory. In modern processors with out-of-order execution, slight variations from strict sequential orderings are allowed to provide speed advantages. A computer program is represented at its lowest level in machine language. Machine language is a representation of instructions as operation codes (opcodes) which are embodied as binary words consisting of zeros and ones. Programs are more conveniently written in a high level programming language such as C and Pascal are even easier. High level programming languages are well known in the art and are used to construct high level instruction sequences wherein each high level instruction may be translated into a collection of machine language instructions.

Most computer systems supply an operating system as an interface between a user's program and a computer's hardware resources. Well known operating systems include UNIX™ as originated from AT&T Bell Labs, and Windows95™ and Windows NT™ as produced by Microsoft Inc. A large body of literature and documentation exists to describe the functionality of operating systems at all levels. An operating system is a computer program which provides driver programs and programs which load and execute other programs. Modern operating systems also typically provide mechanisms to allow different programs to communicate with each other.

When an operating system is present, application programs communicate with each other and with input/output devices using standard high level software interfaces.

These interfaces are commonly known as application programmer's interfaces (API's). An API is accessed by an application through a so-called "function call." A function is a software routine which performs some specified action. A function call is a set of one or more instructions which sends a set of parameters to the function and invokes the function. For example, if a program is written in C and runs on a personal computer running Windows95™, an API may provide a function to allow an application program to

access a protocol stack coupled to an internet. A telephony API may also be supplied to allow an application program to interact with a telephone interface. For example, telephony API functions to pick up a telephone line or to digitize a signal received on a telephone line may be supplied as a function associated with a given API. Operating
5 systems often provide API's as a library functions. An example of an API implemented as a library of functions is the WinSock API which supports Windows95™ and related Windows™ based operating systems. The WinSock API provides roughly fifty function calls, and hence the WinSock system includes a library of fifty functions. It is important to note a given application program may be written to include a call to an API. Also, an
10 operating system may be developed which provides an API as a part of its environment so applications which call the API may execute properly.

Referring now to Figure 7, an application programmer's interface (API) function 700 according to the present invention is illustrated. This function may be included as a part of an operating system and may serve as a gateway between an application process
15 and an operating system process which implements a protocol stack. The inventive API includes two essential inputs. A first input 710 receives information relating to the establishment of an internet socket address, while a second input 720 receives information relating to a PSTN telephone number and/or other CTI related information used for signaling on a CTI link provided by a point-to-point PSTN connection. The API
20 provides access to a protocol stack which may transmit and/or receive data on two separate link layer interfaces. A point-to-point telephone oriented link layer 730 is coupled to a telephone line. An internet layer link interface 740 is coupled via the accessed protocol stack to the internet 110.

The inventive API may be used to provide an interface between an application
25 layer process and a protocol stack in devices such as the smart telephone 115, the CTI server 100, or various versions of the peer devices 102. The API 700 of Figure 7 may be used as a means to program the caller side method 500 or the callee side method 600 previously described. The API 700 is operative to accept both CTI call establishment information as well as internet socket establishment information. This socket
30 establishment aspect of the API 700 provides a functionality associated with a network

interface API. A network interface API provides an interface between an application program process and a protocol stack process which is coupled to an internet. Common network interface APIs include the Sockets API and the WinSock API. The API 700 enables an application or operating system program to interact with a CTI enabled
5 protocol stack used to both set up a point-to-point telephone call and an associated co-socket connection. Utilizing the methods 500, 600 previously described (and variations thereof), the API 700 is operative to transmit internet co-socket establishment data over the point-to-point telephone link layer interface 730, and to perform subsequent co-socket communications over the internet link layer interface 740.

10 Referring now to Figure 7a, a preferred embodiment of a software architecture to be run on a central processing unit (CPU) in a smart telephone or CTI server is presented. An operating system kernel 755 controls various processes which share time on the CPU. The operating system kernel 755 is coupled to an application program 750, a sockets-telephony API 760, and a protocol stack 785. The coupling between the operating system
15 kernel 755 and the application program 750 is implemented as a task control block data structure. The task control block data structure is a data which holds task (i.e. process) related information. Task control block data structures are maintained for one or more resident processes and are manipulated by the operating system kernel 755. As is well known in the art, a process is an instance of a computer program represented by an
20 execution flow. When the process is inactive, a set of process related variables such as a machine state relating to the process are stored in a task control block. The operating system kernel 755 performs task switching by moving the process from a dormant state where it is stored in the task control block to an active state where it is allowed to run on the CPU. By the same token, the coupling between the operating system kernel 755 and
25 the protocol stack 785 is implemented using a second task control block. Depending on the embodiment, the protocol stack may itself be implemented as one or more processes, each of which are coupled to the operating system 755 via their own task control blocks. The operating system kernel 755 is coupled to the sockets-telephony API 760 by virtue of the fact that the sockets telephony API supplies an interprocess communications function
30 which interacts with kernel level data structures and related communication mechanisms.

In some cases the protocol stack 785 may be implemented as a part of an operating system associated with the operating system kernel 755.

The application program 750 is coupled to a co-socket data structure 752. The application program is said to "own" the co-socket 752 when the program makes a
5 function call to create the co-socket. A process which owns the co-socket is able to use the co-socket to communicate with a process on a remote computer. A remote computer may comprise a smart telephone as described hereinabove. The co-socket data structure is embodied within a memory as a collection of information bits. The co-socket data structure includes a standard socket data structure with the possible addition of extra
10 fields for telephone connection related parameters as discussed below. The application program 750 is also coupled to the sockets-telephony API 760. This coupling is implemented within the application program 750 as a function call to a sockets-telephony API function. At run time, the coupling is implemented via a function call which references a function stored in a runtime library. A runtime library is a collection of
15 functions which may be accessed at runtime. The sockets telephony API 760 is coupled to a CTI control module 770 via a coupling 765 which carries telephone connection related information such as connection commands and digitized telephone signals. The CTI control module 770 is coupled to a first hardware driver routine 775. The first hardware driver routine 775 is coupled to a telephone interface 780. The sockets
20 telephony API 760 is also coupled to a protocol stack 785 via a coupling 767 which carries network connection related information such as socket addresses and internet data. The protocol stack 785 is coupled to a second hardware driver routine 790. The second hardware driver routine 790 is coupled to a network interface 795. Depending on the embodiment, a coupling 772 may be implemented to allow the protocol stack 785 to
25 communicate with the CTI control module 770. This is preferably implemented in the form of a function call or via an interprocess communications mechanism supplied by the operating system. When the coupling 772 is implemented, the coupling 765 becomes optional since the sockets-telephony API 760 may pass both socket related and telephone connection related parameters to a transport layer of the protocol stack, and the transport

layer can couple link level information to both the telephone interface 780 and the network interface 795.

It should be noted the telephone interface 780 and the network interface 795 represent link layer interfaces. As discussed above, a link layer interface is defined as a physical layer interface which carries signals, plus a signaling format such as a framing protocol. In a preferred embodiment the physical layer interfaces of the telephone interface 780 and the network interface 795 are different. In some embodiments, such as those involving an integrated services digital network (ISDN) line, a DSL, or a T1 line for example, the two link layer interfaces 780 and 795 may be carried on the same physical connection. In the preferred embodiment, the telephone interface 780 is coupled directly to a remote computer via a point-to-point telephone connection, and the network interface 795 is coupled to the remote computer via an indirect connection through an internet.

The sockets-telephony API function module 760 itself includes structural features. The module 760 includes a first software sub-module which is coupled to the CTI control interface 770 which is itself operably connected to the telephone interface 780. The module 760 also includes a second software sub-module coupled to the first software sub-module. The second software sub-module is coupled to the co-socket data structure 752. Preferably, this coupling is implemented by having the application program 750 pass a pointer to the sockets-telephony function 760. The module 760 may optionally also include a connection to the protocol stack 785.

Next consider the operation of the software architecture as illustrated in FIG. 7a. Upon application of power to the system, the operating system is loaded into memory at boot-time from a disk. In alternative embedded embodiments, the operating system may be resident in a semiconductor memory such as ROM when power is turned off, in which case the operating system is not loaded from disk. Once the operating system is loaded, control passes to the operating system kernel 755. The operating system kernel loads various processes to run such as the application program 750 and the protocol stack 785. The application program 750 makes function calls to the sockets-telephony API 760 in order to establish a communication socket with a remote computer. For example, using



well known concepts of UNIX™ sockets and Windows™ Sockets (WinSock) API's, a function called "socket" is called to create a socket data structure. The socket function returns a pointer to the created socket data structure. A socket data structure is an endpoint for communication and provides a means to allow the application 760 to send and receive messages with a remote computer. An alternative function called "co-socket" may be used create and return a pointer to the co-socket 752. The co-socket 752 contains the same information as a socket data structure plus information relating to a telephone connection. A process which makes a function call to the co-socket function is said to "own" the co-socket. The function owning the co-socket is able to use the co-socket for as a data structure in support of network communication. Further details describing the use of the co-socket data structure 752 are discussed immediately below.

Consider the case when the application program 750 wishes to establish a connection with a remote computer. Once the application program 750 has obtained a socket or co-socket data structure as discussed above, the application program 750 calls a "connect" function. A connect function, as is also known in the sockets literature, is a software function which accepts as an argument a pointer to a socket data structure, an address of a remote socket, and a length of the remote socket address. In accordance with the present invention, a new type of connect function called a "co-socket connect" function is supplied which takes an additional argument of a telephone number. In a preferred embodiment, the co-socket connect function according to the present invention only requires a pointer to a socket data structure and a telephone number of the remote computer to be sent as input parameters. In an alternative preferred embodiment, the co-socket connect function only requires a pointer to the co-socket data structure 752 as an input parameter. The co-socket data structure 752 then contains the same information as a standard socket data structure plus a field relating to a telephone connection. For example, the field may contain a telephone number to be dialed to establish a connection with a remote smart telephone or CTI server. Once the application program 750 calls the co-socket connect function as provided by the sockets-telephony API 760, the co-socket connect function is operative to direct a data segment to be sent to the remote computer by dialing the telephone number of the remote computer and sending a data segment to



the remote computer via the dialed telephone connection. Preferably, the data segment is a SYN segment used in the establishment of a TCP/IP stream socket in accordance with the method 600. In that case, a SYN segment is routed over the telephone interface 780 via a point-to-point connection to the remote computer, and a TCP/IP co-socket is established to support the call using the network interface 795. In accordance with the present invention, the sockets-telephony API 760 need only be aware of the remote computer's telephone number in order to establish an internet co-socket connection with the remote computer. In practice, this sequence of events may be initiated when a caller using the smart telephone 115 dials a phone number which is answered by the CTI server

10 100.

Next consider the case where the application program 750 is configured to receive incoming phone calls. For example, a CTI server application program may be designed which receives incoming telephone calls from clients and sets up co-sockets in response thereto. In this case the application program 750 calls a socket function to establish a socket data structure or the co-socket data structure 752. As discussed above, the co-socket data structure 752 differs from a standard socket data structure in that it holds additional information relating to a telephone connection. Also, the co-socket 752 be a communication end point for a co-socket established at least partially over a point-to-point telephone connection. Next the application program 750 calls a co-socket listen function. Listen functions are well known in the sockets art and are used by servers to listen for client requests over a socket connection. In accordance with the present invention, the sockets-telephony API 760 includes a co-socket listen function which listens for incoming telephone calls carrying co-socket establishment segments. When a telephone call is received over the telephone interface 780, the listen function is triggered into an active state. When a call is received, the listen function causes the CTI control module 770 to process an incoming signal to determine whether a SYN data segment has been received. If a SYN data segment has been received, the CTI control module 770 passes the SYN segment to the protocol stack 785 via the coupling 772. The protocol stack 780 thus uses the telephone connection as a second link layer interface over which to receive incoming SYN segments. While it is anticipated that other types information

15
20
25
30

may advantageously be passed over the telephone connection to the protocol stack, the transmission of SYN segments is viewed as the preferred use. Hence the listen function is operative to receive incoming phone calls and to set up co-sockets across an internet in response thereto, just as a conventional listen function does using only the network interface 795. As discussed above, similar structures using session layer interfaces are also within the scope of the invention. These embodiments may use session layer packets different from SYN segments and session-layer versions of a co-socket data structure.

The software architecture of Figure 7a illustrates an aspect of the present invention relating to the protocol stack 785. The protocol stack 785 includes the coupling 772 to the telephone connection oriented link layer interface 780 and a coupling to the network connection oriented link layer interface 795. The protocol stack may be passed a first input relating to a socket data structure and a second input relating to a telephone connection. When implementing a connect function, the second input preferably includes the telephone number of the remote computer to which the protocol stack is directed to connect. Alternatively, the first and second inputs may be embodied within in the co-socket data structure 752. In operation, the protocol stack 785 receives a pointer to the co-socket 752 as an input parameter. The co-socket 752 contains the first and second inputs. Next the protocol stack causes the telephone number to be dialed, and then transmits a SYN segment over the established point-to-point telephone connection. The method 600 is preferably employed by the protocol stack to establish a co-socket connection with the remote computer. Alternatively, the telephone number may be directed to be dialed using the coupling 765, in which case the co-socket data structure 752 need only contain a telephone connection identifier. The protocol stack thus transmits the SYN segment over the designated connection 780 and preferably implements the method 600. Once the co-socket is established, subsequent internet communication occurs over the network connection 795.

When implementing a listen function, the second input preferably includes a telephone line identifier for one or more telephone lines over which telephone calls requiring co-socket services may be received. The co-socket listen function is passed a pointer to the co-socket 752 which includes a socket data structure plus a telephone line

identifier. When the designated telephone line receives an incoming call which carries a SYN segment, the CTI control module 770 signals the event to the protocol stack 785 via the coupling 772. The protocol stack 785 is operative set up a co-socket for subsequent communication over the network connection 795. This connection is set up in response to the SYN segment received on the telephone interface 790. The protocol stack 785 preferably implements the method 500 to establish the co-socket connection with the remote computer in response to the incoming call. If the coupling 765 is used, the application program may be involved in routing the received SYN segment back to the protocol stack via the coupling 767.

Figure 8 illustrates a filter program according to the present invention which enables users to, inter alia, share images and other forms of computer data over a co-socket internet connection while conversing over a PSTN telephone connection. A filter routine 800 is coupled via software connections to a first process 805 and a second process 810. The filter program is also coupled to a sockets API 820 which provides access to a protocol stack 822. The process 805 may receive input from various sources to include a sensor device 815. The sensor device may include a digital camera, a keyboard, a mouse, and/or any other devices which may be connected to receive input. Resident in the same computer is a "call initiate" routine 825 which is coupled to a CTI API 830. The CTI API 830 may optionally receive an input from a connection establishment routine 840. The connection establishment routine 840 is coupled to provide input and make calls to the CTI API 830 and the sockets API 820.

The inventive filter is operative to intercept at least a portion of a data stream sent from the first process 805 to the second process 810 and to route the intercepted data across a co-socket maintained by the protocol stack 822. It is understood that the filter 800 may be coupled directly to the protocol stack 822 without using an API, but an API represents a preferable approach. To understand the function of the filter 800, consider the following example. Assume that the first process 805 is a word processing program. Assume that the second process 810 is a windows process which receives data from the first process and displays the received data in a window on a local computer display screen. In such a situation, the filter routine 800 copies this data stream and preferably

formats it into an application layer packet stream suitable for transmission to an application layer program which resides on a remote computer coupled to an internet co-socket connection as managed by the protocol stack 822. In a preferred embodiment, the co-socket is established using the method 500 or the method 600. However, the CTI API 830 and the sockets API 820 may also be merged using functions similar to the function 700.

The preferred use of the filter 800 is to allow remote users to converse on a communications device (such as a telephone) while sharing computer data via an internet co-socket. Referring again to the previous example, remote colleagues may be discussing a document over the telephone. Since the filter program intercepts and copies the data stream sent from a word processor to a windows display process, the copied data stream will contain display information needed to display the word processor data on the remote computer's screen. Hence the filter sends this data across the co-socket to the remote site so the colleague on the remote end of the phone line may view and discuss the same word processor display window. As discussed previously, this type of data and media sharing is known as data conferencing. Hence the filter 700 enables a mixed mode of data conferencing whereby preferably voice is carried over a toll quality PSTN circuit and other forms of media and data are transported across an internet via a co-socket.

Figure 9 illustrates a method of processing 900 carried out within a computer or smart phone. In a first step 910 a first process transmits a data stream to a second process. In some embodiments, the data stream may be transmitted from a first thread to a second thread, from a first module to a second module, or most generally from the domain of a first set of instructions to the domain of a second set of instructions. As is well known in the art, a "thread" is an execution flow which may exist independently from others. In many computer systems, a process may have one or more threads of execution. A second step 920 is preferably implemented using a third process separate from the first and second processes. This third process is a preferably a filter process interposed between the first process and the second process. Control in the filter process next passes to a third step 930 which packetizes the intercepted information stream. The packetized data is preferably packetized into an application layer packet stream formatted

for use by an application program residing on a remote computer (which is coupled via a co-socket). Control next passes to a fourth step 940 which sends the packetized data stream to the co-socket. This method enables real-time data conferencing to occur by sending voice data over a PSTN voice circuit and image or other forms of computer data across an internet co-socket.

Figure 10 illustrates an exemplary central office switching arrangement 1000 in accordance with the present invention. A subscriber line interface circuit module is preferably coupled to fiber and cables carrying large numbers of twisted pairs which provide subscribers with local loop access to the PSTN. The signals carried on this wiring are typically digitized and multiplexed and coupled to a digital central office (DCO) 1020. The digital central office 1020 includes a stored program computer (SPC) module 1030 which executes a switch generic program providing digital control, network management, and other services. The SPC module 1030 is coupled to an automatic number identification/automatic socket identification (ANI/ASI) data base and translation module 1040. The ANI/ASI database and translation module 1040 may be a part of the DCO and coupled to mass storage devices (not shown). Additionally, the ANI/ASI database and translation module may optionally be coupled to a TCP/IP protocol stack 1050, preferably via a sockets API (not shown). The DCO preferably includes a main communication data path and a signaling path which couples to a switching fabric 1060. The switching fabric 1060 couples to an interoffice trunk interface module 1070 which couples long distance traffic out of the local access transport area (LATA).

The central office switching arrangement 1000 of Figure 10 is operative to switch intra-LATA traffic as well as inter-LATA traffic as is well known in the art. The central office switching arrangement preferably uses a standard SPC DCO architecture which is operative to perform time division multiplexing of subscriber channels and to provide services such as ANI/CLID, automatic message accounting, and maintenance and control functions. The switching fabric 1060 may be implemented using any one of a number of well understood switching fabric architectures such as space-time-space, time-space-time, or shared memory ATM-packet, for example. Since central office switching is a mature technology and well known to those of ordinary skill in the telephonic arts, these aspects

will not be discussed further herein. Of note, however, is the ANI/ASI database and translation module 1040 associated with the central office switch in the present invention.

The ANI/ASI module is operative to translate ANI or related CLID information available to the DCO 1020 into internet socket address information. This translation is carried out in the present embodiment statistically by sending a database query to a mass storage unit which uses the ANI or CLID data as a key and returns an internet socket address. At any point herein, the term "ANI" may be interpreted to mean "ANI or CLID" data. Typically ANI data is used within a telecommunications network for billing while a CLID packet is delivered to the user. The issue of importance for use herein is that ANI and CLID both to carry information indicative of the identity of the caller. The process of converting ANI data to an internet socket address is called ASI, for "automatic socket identification." The translation is preferably carried out dynamically by maintaining socket connections to users or connection servers via an internet coupled via the protocol stack 1050, although other approaches are possible. When a number is to be checked, a dynamic database is accessed which keeps track of users currently connected with active internet connections. In the dynamic translation scenario, when a user logs onto an internet connection, an application program resident on the user's computer or ISP automatically establishes a socket connection for co-socket use and forwards this socket number to an internet address associated with the central office 1000. The central office 1000 performs the ANI/ASI translation and preferably either replaces CLID information in PSTN data packets with ASI data, or else appends the ASI data to the CLID data so both the caller's number and internet socket address are available to the called end of the telephone connection.

In most systems, the PSTN data packets correspond to SS7 packets. SS7 signaling, as is well known, uses common channel signaling to perform call set-up and call termination. This system differs from prior art systems in that the database translation is used to identify an active internet port address such as used in a PPP connection to allow the callee to return the call by popping a screen of information on the caller's computer screen using the co-socket address information found in the database translation. While prior art systems provided static information such as e-mail, FTP, and

web-page URL information within CLID packets, the present invention provides socket port numbers to enable real-time multimedia communications via co-sockets. It should be noted that while the switching arrangement 1000 has been discussed in connection with a central office switch within the PSTN, the same architecture may be applied to the design of a PBX. In general, a telephone switch may be a PBX, central office switch, long distance switch, or a switch involving internet telephony used to switch internet calls. The apparatus 1000 may be applied in general to a telephone switch.

Figure 11 illustrates a method 1100 practiced primarily by a local switching office employing the central office switching arrangement 1000. This method may also be practiced more generally by a telephone switch. In a first step 1110, a PSTN data packet is received containing ANI or CLID information. This data packet is preferably a call-set-up related data packet which is received before the actual voice circuit is established. In some cases the received data packet uses an SS7 structure, and in other cases it uses a different structure for the particular central office in use. After the PSTN data packet is received containing the caller's identification information in the first step 1110, control passes to a second step 1120 wherein a database query is performed to associate the caller's identification information with an internet address which is preferably a TCP/IP stream socket or datagram socket address. As discussed previously, the database may be a static database maintained in a mass storage device, or may preferably be a dynamic database updated each time a user opens an active client internet session. An active client internet session may be, for example registered by an Internet server when a user has connected over a PPP connection. If no match is found associating the caller's number to an internet address, control passes from the second step 1120 to a third step 1130. In the third step 1130, a call set-up packet is preferably forwarded to the calling end containing only standard CLID information.

If a match is found in the second step 1120, control may pass to either a fourth step 1140 or a sixth step 1160. The decision as to which of the steps 1140 or 1160 is carried out is set according to a state variable SV1 which may represent a software variable setting or a hardwired setting. That is, in some embodiments, only one of the steps 1140 or 1160 may be available. If a match is found in the second step 1120 and the

state variable SV1 is in a first position, control passes from the second step 1120 to the fourth step 1140. In the fourth step 1140, an internet socket address is inserted into a PSTN data packet for transmission to the called party. Control next passes from the fourth step 1140 to a fifth step 1150 which forwards the PSTN data packet for transmission through the PSTN. The PSTN data packet is preferably an SS7 call set-up packet carried over a common channel using common channel signaling. If a match is found in the second step 1120 and the state variable SV1 is in a second position, control passes from the second step 1120 to the sixth step 1160. In the sixth step 1160, a data packet is sent to the caller via an internet connection which is preferably the socket found in the ANI/ASI translation database step 1120. Depending on a second state variable SV2 control may pass to different places after the step 1160. The state variable SV2 may represent a software bit field or a hardware setting depending on the implementation. If the state variable SV2 has a first value, no further action is taken and the call set up process is terminated as illustrated in the termination oval 1170. If the state variable SV2 has a second value, control is passed to step the 1140 which behaves as discussed above. If the state variable SV2 has a third value, control passes to the fifth step 1150 which passes a PSTN data packet with no internet address information supplied.

Consider now some practical applications of the method 1100. Suppose a caller is to call a 1-800/888 phone number. Suppose the caller is calling from either a smart phone 115 or from an office with a separate phone line and internet connection. Suppose the caller also has registered for ANI/ASI service with the LEC. When the user places a call, a socket address is found in the database query step 1120. If the state variables are configured to pass control to the fourth step 1140, the caller's co-socket address will be provided in a PSTN call-set up packet which will be carried to the called 1-800/888 phone number's location. When the phone rings at the remote location, the caller's co-socket will preferably be made available along with CLID information. CLID information is delivered in a CLID packet which may be a part of an SS7 packet, some other form of PSTN data packet, or an emerging standard's internet telephony packet. At this point the remote 1-800/888 service provider may display a screen of information on the caller's computer and optionally reject the call. This saves the 1-800/888 service



provider money and gives the caller a more friendly user interface than IVR voice prompts. The call center also may provide a web-call back timer according to the aspects of the present invention previously discussed in connection with Figure 2, or may be used for peer-to-peer multimedia co-socket communication. If the state variables are configured to direct control to pass to the sixth step 1160, the ANI/ASI translation system may provide the called number's internet address back to the caller. In this way, the caller's computer may initiate a session with the called phone number's server. No PSTN packet need be forwarded across the PSTN in this case. In this mode of operation, the database query step 1120 must also locate the called number's internet address in the database.

Figure 12 illustrates an arrangement whereby the method 1100 and related methods may be implemented in a IXC network configuration 1200. A LEC 1210 is viewed by the IXC network 1200 as a signaling end point. The LEC is coupled to transfer SS7 call set-up and call termination packets to a tandem switch 1220. The tandem switch is an IXC long distance switch, also known as a class 4 switch. The tandem switch 1220 is coupled via an SS7 link to an adjunct processor (AP) 1230. The tandem switch 1220 may include an embedded service switching point which processes calls requiring database translations. The tandem switch 1220 is also coupled via an SS7 link to a signal transfer point (STP) 1240. An STP is a signaling node which acts as a hub for SS7 signaling messages. The STP is linked via an SS7 link to a service control point (SCP) 1250 which supplies database information to a set of network nodes. The network is controlled by an operations system 1260 which provides network management and other capabilities. The operations system 1260 is coupled normally by X.25 data links to the adjunct processor 1230 and the service control point 1250. In the embodiment shown, the adjunct processor 1230 contains an adjunct service point which responds to requests for service processing. This adjunct service point is coupled to a CLID-to/from internet address translation unit 1240 to provide automatic socket identification (ASI). The translation unit 1240 may optionally be coupled to an internet 1250 to provide a dynamic database of current internet addresses available for use as co-

sockets and related purposes. In some embodiments the translation unit 1240 may be coupled to the SCP 1250.

Consider the operation of the network configuration 1200. A caller accesses the LEC 1210 via a subscriber loop. Supposing a long distance call is placed, an SS7 call set-up packet is routed from the LEC to the IXC. In which case, the LEC is viewed by the network 1200 as a signaling end point. When the SS7 packet reaches the tandem switch 1220, it is routed through an internal service switching point. In the illustrative embodiment, if the call is identified as requiring internet co-socket database translation services, the SSP may route at least part of the SS7 packet to either the adjunct processor 1230 or the service control point 1250 for database translation. In Figure 12, the packet is routed to the adjunct processor 1230 for translation. In this embodiment, the IXC network node may implement the method 1100 or similar methods to provide internet co-socket information to a called end of a connection. Data may transfer from the common channel signaling SS7 network to the internet 1250 to facilitate the establishment of co-sockets using a dynamic database as discussed in connection with Figure 10 and Figure 11 above.

Referring now to Figure 13, a method 1300 is illustrated in flow chart form for establishing a co-socket without the need to transmit SYN segments over a telephone network. This method is practiced by the smart telephone 115 in the present embodiment; however, it can be appreciated that other types of devices may utilize the method with equal success. In a first step 1310, a phone number is dialed or retrieved for automatic dialing from a calling list. Control is then passed to a second step 1320 which performs a database translation to convert the telephone number into an internet address. As illustrated in Figure 1, a database used to perform the translation is located in the local storage unit 165. Alternatively, the database may be located in the remote internet database server 112, or other locations based on the specific application. Depending on the information returned by the database search, a state variable (SV) is set. In some embodiments this state variable may be hard-wired, or different control flow logic may be employed. In the illustrative embodiment of the method 1300, depending on the value of the state variable, control passes from the second step 1320 to either a third step 1330, a

fourth step 1340, or a sixth step 1360. If SV is in a first position, control passes from the second step 1320 to the third step 1330. In the third step 1330, a telephone number is dialed to establish a point-to-point PSTN or PBX telephone connection with the called party. Control then passes from the third step 1330 to a seventh step 1370 where
5 communication with the distant end occurs. In this case communication is via the point-to-point PSTN or PBX telephone connection established in the third step 1330. This control path is typically executed if no database match is found or if the database supplies information indicating no co-socket address is available for the called number.

Alternatively, if the state variable is in a second position, control passes from the
10 second step 1320 to the fourth step 1340. In the fourth step 1340, a telephone number is dialed to establish a point-to-point telephone PSTN connection with the callee. Control passes from the fourth step 1340 to the fifth step 1350 where an internet co-socket is established by sending SYN segments across an internet. Note that the fourth step 1340 and the fifth step 1350 may be executed in any sequence or in parallel. Control now
15 passes from the fifth step 1350 to a seventh step 1370 where communication with the distant end occurs. In this case communication is via both a point-to-point PSTN telephone connection and an internet co-socket. This control path is useful for establishing multimedia links using a point-to-point PSTN telephone connection and an internet co-socket.

20 If SV is in a third position, control passes from the second step 1320 to the sixth step 1360. In the sixth step 1360, an internet connection is established by sending SYN segments across an internet. Alternatively, datagrams may be sent without establishing a TCP stream socket. Control passes from the sixth step 1360 to the seventh step 1370 where communication with the distant end occurs. In this case communication is via an
25 internet connection only. This control path is useful for establishing an internet link to an internet call center such as the call center 140 accessible via the CTI server 100. It may also be used to reach a web page by dialing a telephone number from a smart phone. If the number is dialed from a non-smart phone a CTI server may be reached. If the number is dialed from a smart phone, a graphical interface can be reached. The method 200
30 previously discussed may be used to obtain a call-back from the call center at a later time

in this form of co-socket telephony. No voice connection is initially needed. An alternative use is to establish an H.323 link if the database translates the PSTN telephone number into a packetized connection telephone number.

Figure 14 illustrates a method 1400 of removing internet socket address information from an SS7 call set-up packet. In a first step 1410, a PSTN data packet is received. For example, this step may be carried out in a switching service point within the tandem switch 1220. Control next passes to a second step 1420 which preferably checks the PSTN data packet for internet address information embedded into the CLID portion of the packet. If no internet address information is found, control transfers to a third step 1430. The third step 1430 forwards the PSTN data packet for further processing and transmission. Alternatively, if an internet address is found in the second step 1420, control passes to a fourth step 1440 which is operative to remove the internet information from the data packet. This fourth step may be performed by adjunct processor 1230, for example, built around an Intel Pentium™ processor or other such device. Once the internet address information is removed, control transfers to a fifth step 1450 which is operative to forward the packet for further processing and transmission. This method is advantageous to an IXC whom does not want to pass traffic which may be useful to internet competitors.

Although the present invention has been described with reference to specific embodiments, other embodiments may occur to those skilled in the art without deviating from the intended scope. For example, any of the methods disclosed herein may be modified using different control variables and/or different sequencing of steps while achieving similar results. Also, while certain methods were described in the context of a PSTN, they may also be practiced using Internet telephony connections. Additionally, any number of different hardware, firmware, and software combinations may be employed to embody the apparatus disclosed herein. Therefore, it is to be understood that the invention herein encompasses all such embodiments which do not depart from the spirit and scope of the invention as defined in the appended claims.